Documentation for superluminal_demo_maxwell.py

This Python script simulates the propagation of electromagnetic waves using Maxwell's equations. It utilizes numerical methods to calculate the electric and magnetic fields over a specified number of time steps.

Overview of the Code Structure

The code is structured to initialize the necessary parameters for the simulation, compute the propagation of the fields over discrete time steps, and visualize the results using matplotlib.

Variables

`num_timesteps`

Defines the total number of time steps for the simulation. This is set to 400.

`n`

Specifies the size of the grid for the simulation. Here, it is set to 1000, which means the simulation will work on a 1000x1000 grid.

`epsilon`

The permittivity of free space, defined as `8.85 * 10^(-12)`, a constant used in electromagnetic calculations.

`mew`

The permeability of free space, calculated as `4 * np.pi * 10^(-7)`, another constant for electromagnetic equations.

`Ez`

A 2D array initialized to store the electric field values in the z-direction. Its size is `n x n`.

`Hy`

A 2D array initialized to store the magnetic field values in the y-direction, sized to `(n-1) x (n-2)`.

`Hx`

A 2D array initialized to store the magnetic field values in the x-direction, sized to `(n-2) x (n-1)`.

`c`

The speed of light in the medium, computed using the formula `1/sqrt(epsilon*mew)`.

`S`

Stability factor used in the time-stepping algorithm, calculated as `1 / sqrt(2)`.

`dx`

The spatial step size, set to `1.0 * 10^(-6)` meters.

`dt`

The time step size, computed as `S * dx / c`.

`lambd`

The wavelength, set to `20.0 * dx`.

`omega`

The angular frequency, calculated using `2 * np.pi * c / lambd`.

`Ca`, `Da`

Constants for updating the electric field, both set to `1.0`.

`Cb`, `Db`

Constants for updating the magnetic field, calculated using the respective formulas involving `dt`, `epsilon`, and `mew`.

`temp`

A temporary array used to store the electric field values for plotting. Its size is `1000 x n`.

`j`

Counter variable initialized to zero, used for indexing during the simulation.

Processing Loop

The main loop of the code runs for the specified `num_timesteps`. During each iteration:

1. If it's the first time step, the electric field `Ez` at the center of the grid is set to 1.
2. The magnetic fields `Hy` and `Hx` are updated based on the current electric field values.
3. The electric field `Ez` is updated using the newly calculated magnetic fields.
4. Prints the current time step for tracking progress.

Visualization

After the processing loop, the script prepares to visualize the results:

1. It initializes a temporary array `temp` to store the diagonal values of the electric field `Ez`.
2. The x-axis is set up to represent the radial distance.
3. The results are plotted using `matplotlib`, with labels and limits set for clarity.
4. Finally, the plot is displayed to the user.

Note: Some parts of the visualization code are commented out, which means they won't execute as is. Uncomment them if you wish to animate the results!

Conclusion

This script provides a basic framework for simulating electromagnetic wave propagation. By adjusting parameters like `num_timesteps` and grid size `n`, users can explore various scenarios in wave behavior.